Compressive Sensing

Dr.-Ing. Ali Bereyhi

Friedrich-Alexander University

ali.bereyhi@fau.de

Summer 2022

Welcome to the Course *Compressive Sensing*

When and where?

- Time: *Tuesdays 14:15 – 15:45 and Wednesdays 16:15 – 17:45*

  None of the sessions lies on a holiday!

- Place: *Lecture Room 05.025, 5th Floor, Cauerstraße 7*

How do we proceed?

- Handouts are uploaded prior to each session on StudOn
- We have the sessions *in person*
- Further materials are provided through StudOn

Welcome to the Course *Compressive Sensing*

What are the materials for this course?

- *You receive handouts for each part of the course*
- *There are some tutorial assignments and homework*
  - *You are free to submit these assignments*
  - *At the end of the semester, you can choose whether*

    your submissions impact your grade or not
- *The final exam will be Oral*

Welcome to the Course *Compressive Sensing*

What are the textbooks?

- *A Mathematical Introduction to Compressive Sensing*
  S. Foucart and H. Rauhut, *Book*, 2013.
- *Statistical Mechanics of Regularized Least Squares*
  A. Bereyhi, *PhD Dissertation*, 2020.

How to have access to lecture notes?

- *You could write your own lecture notes*
- *We could also appoint a writer for each session, if you wish*

Welcome to the Course *Compressive Sensing*

Any questions? *Simply contact me!*

- Ali Bereyhi
  - *Room 04.021, Cauerstraße 7, 91058, Erlangen*
  - `ali.bereyhi@fau.de`

An Overall Look

What We Learn in This Course

What is our final goal?

Learn sparse recovery and be able to use it

But, what is sparse recovery?

We learn it in *Part Zero* of the lecture

Part Zero: Introduction to Compressive Sensing

- The Problem of Sparse Recovery
- Sample Applications of Sparse Recovery

What We Learn in This Course

Now that we know the problem, how do we do sparse recovery?

We learn it in *Part One* of the lecture

Part One: Compressive Sensing from the Classical Viewpoint

- Our First Try for Noise-free Sparse Recovery
- Recovering Sparse Signals from Noisy Measurements
- Good Sensing Matrices for Compressive Sensing
- Performance Guarantees for Sparse Recovery Algorithms

What We Learn in This Course

But, is sparse recovery all about classical sensing systems?

No! We see this in *Part Two* of the lecture

Part Two: Compressive Sensing from a Bayesian Viewpoint

- Formulating Sparse Recovery as a Bayesian Inference
- Minimum Mean Squared Error Bound for Sparse Recovery
- Regularized Least-squares as a Bayesian Recovery Algorithm
- Introduction to Approximate Message Passing

Recovering Sparse Signals

An Illustrative Example

Consider the following system of linear equations:

$$
\begin{aligned}
x_1 + x_2 + x_3 &= 3 \\
x_1 - x_2 + 3\, x_3 &= 1 \\
x_3 + x_4 &= 1
\end{aligned}
$$

Answer the following items:

(a) How many solutions does this system of equations have?

An Illustrative Example: *Answer to Part (a)*

For a system of linear equations with

- *N unknowns, and*
- *M linearly independent equations*

we have the following cases:

- *Overdetermined: When $N < M$*
  - *We have no solution*
- *Determined: When $N = M$*
  - *We have a single solution*
- *Underdetermined: When $N > M$*
  - *We have multiple solutions*

An Illustrative Example: *Answer to Part (a)*

In this example, we have

- $N = 4$ *unknowns:* x_1, x_2, x_3 *and* x_4
- $M = 3$ *linearly independent equations: For any* α, β *or* γ

$$x_1 + x_2 + x_3 = 3 \qquad [1,1,1,0] \neq \alpha\,[1,-1,3,0]$$
$$x_1 - x_2 + 3\,x_3 = 1 \qquad [1,1,1,0] \neq \beta\,[0,0,1,1]$$
$$x_3 + x_4 = 1 \qquad [1,-1,3,0] \neq \gamma\,[0,0,1,1]$$

Thus, we conclude that

Underdetermined: Since $N = 4 > M = 3$

- *We have multiple solutions*

An Illustrative Example

Consider the following system of linear equations:

$$
\begin{aligned}
x_1 + x_2 + x_3 &= 3 \\
x_1 - x_2 + 3\, x_3 &= 1 \\
x_3 + x_4 &= 1
\end{aligned}
$$

Answer the following items:

(b) Calculate the solutions of this system of equations.

An Illustrative Example: *Answer to Part (b)*

Let $x_3 = u$ for a real scalar u

- *From the third equation, we have*

$$u + x_4 = 1 \rightsquigarrow x_4 = 1 - u$$

- *From the second equation, we have*

$$x_1 - x_2 + 3u = 1 \rightsquigarrow x_1 = 1 + x_2 - 3u$$

- *Replacing x_1 in the first equation results in*

$$1 + x_2 - 3u + x_2 + u = 3 \rightsquigarrow x_2 = 1 + u$$
$$\rightsquigarrow x_1 = 2 - 2u$$

An Illustrative Example: *Answer to Part (b)*

Thus, for any real scalar u, the vector

$$\boldsymbol{x}(u) = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 2-2u \\ 1+u \\ u \\ 1-u \end{bmatrix}$$

is a solution to the system of equations

An Illustrative Example

Consider the following system of linear equations:

$$
\begin{aligned}
x_1 + x_2 + x_3 &= 3 \\
x_1 - x_2 + 3\, x_3 &= 1 \\
x_3 + x_4 &= 1
\end{aligned}
$$

Answer the following items:

(c) A side constraint tells us

At most two of the unknowns are non-zero

What are the solutions which satisfy this constraint?

An Illustrative Example: *Answer to Part (c)*

What are the solutions which have zero entry?

- *For $u = 1$, $u = -1$ and $u = 0$, we have*

$$\boldsymbol{x}(u = 1) = \begin{bmatrix} 0 \\ 2 \\ 1 \\ 0 \end{bmatrix} \quad \boldsymbol{x}(u = -1) = \begin{bmatrix} 4 \\ 0 \\ -1 \\ 2 \end{bmatrix} \quad \boldsymbol{x}(u = 0) = \begin{bmatrix} 2 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

- *For any other real u, we $\boldsymbol{x}(u)$ has no zero entry*

An Illustrative Example: *Answer to Part (c)*

Thus, we conclude that there is only one solution which

has at most two non-zero entries

This solution is

$$\boldsymbol{x}\left(u = 1\right) = \begin{bmatrix} 0 \\ 2 \\ 1 \\ 0 \end{bmatrix}$$

An Illustrative Example

Moral of Story ...

- *An underdetermined set of equations has multiple solutions*
- *A side constraint can lead to a unique solution*

Recovering Sparse Signals

Section 1: The Problem of Sparse Recovery

Sparse Recovery

Sparse Recovery

In the problem sparse recovery, we intend to find a sparse solution of an underdetermined system of equations

Immediately, the following questions come to your mind:

- *What does sparse mean?*
- *Why are we talking about sparse recovery?*
- *Why haven't we yet talked about compressive sensing?*

No worries! These questions get clear shortly

Sparse Recovery: *Basic Definitions*

The system of linear equations is further represented as

$$\mathbf{A}\boldsymbol{x} = \boldsymbol{y}$$

where $\mathbf{A}$, $\boldsymbol{x}$ *and* $\boldsymbol{y}$ *are*

- $\boldsymbol{x}$ *is an* N*-dimensional vector of unknowns*
- $\mathbf{A}$ *is an* $M \times N$ *matrix*
- $\boldsymbol{y}$ *is an* M*-dimensional vector*

Sparse Recovery: *Basic Definitions*

The system of linear equations is further represented as

$$\mathbf{A}\boldsymbol{x} = \boldsymbol{y}$$

Back to our first example

$$x_1 + x_2 + x_3 = 3$$
$$x_1 - x_2 + 3\,x_3 = 1$$
$$x_3 + x_4 = 1$$

Sparse Recovery: *Basic Definitions*

The system of linear equations is further represented as

$$\mathbf{A}\boldsymbol{x} = \boldsymbol{y}$$

Back to our first example

$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & -1 & 3 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix}$$

Sparse Recovery: *Basic Definitions*

The system of linear equations is further represented as

$$\mathbf{A}\boldsymbol{x} = \boldsymbol{y}$$

Remember that for underdetermined systems

$\mathbf{A}$ *is a fat matrix*

I.e., it has more columns than rows

Sparse Recovery: *Basic Definitions*

What is the support of a vector?

- *The indices of non-zero entries of the vector, e.g.,*

$$\boldsymbol{x} = \begin{bmatrix} 5 \\ 0 \\ 4 \\ 0 \end{bmatrix} \rightsquigarrow \mathrm{Supp}\left(\boldsymbol{x}\right) = \{1, 3\}$$

What is an s-sparse vector?

- *A vector whose support has s elements, e.g.,*

  $\boldsymbol{x}$ *is 2-sparse*

Sparse Recovery: *Basic Definitions*

The sparsity of a vector is given by the ℓ_0-norm of the vector

For the vector $\boldsymbol{x}$ with N entries, the ℓ_0-norm is defined as

$$\|\boldsymbol{x}\|_0 = \sum_{n=1}^{N} \mathbf{1}\left\{x_n \neq 0\right\}$$

where $\mathbf{1}\left\{\cdot\right\}$ is the indicator function

$$\mathbf{1}\left\{X\right\} = \begin{cases} 1 & \text{if argument } X \text{ holds} \\ 0 & \text{if argument } X \text{ does not hold} \end{cases}$$

Sparse Recovery: *Basic Definitions*

The sparsity of a vector is given by the ℓ_0-norm of the vector

Get back to the example

$$\boldsymbol{x} = \begin{bmatrix} 5 \\ 0 \\ 4 \\ 0 \end{bmatrix}$$

The ℓ_0-norm is

$$\begin{aligned} \|\boldsymbol{x}\|_0 &= \mathbf{1}\{x_1 \neq 0\} + \mathbf{1}\{x_2 \neq 0\} + \mathbf{1}\{x_3 \neq 0\} + \mathbf{1}\{x_4 \neq 0\} \\ &= \mathbf{1}\{5 \neq 0\} + \mathbf{1}\{0 \neq 0\} + \mathbf{1}\{4 \neq 0\} + \mathbf{1}\{0 \neq 0\} \\ &= 1 + 0 + 1 + 0 = 2 = \textit{sparsity of } \boldsymbol{x} \end{aligned}$$

Sparse Recovery: *Basic Definitions*

How could we formulate sparse recovery mathematically?

Sparse Recovery

For given $\mathbf{A}$ *and* $\boldsymbol{y}$, *we intend to find vector* $\boldsymbol{x}$, *such that*

$$\mathbf{A}\boldsymbol{x} = \boldsymbol{y}$$
$$\text{subject to } \|\boldsymbol{x}\|_0 \leq s$$

for some sparsity s

Shouldn't I write "subject to $\|\boldsymbol{x}\|_0 = s$*"?*

$\boxed{\textit{You get an assignment which makes it clear!}}$

Sparse Recovery

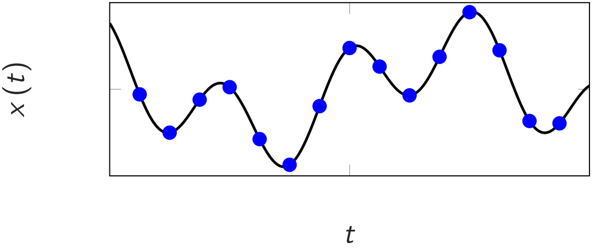
A typical audience would have this conversation with me now:

+ *Nice! It is good to know what does sparse recovery mean! But it is purely mathematical!*
– *Somehow yes!*
+ *What is the connection to the so-called compressive sensing?*
– *Well, it is literally what we do in compressive sensing*
+ *OK! Then, what does it have to do with someone who works on communications and signal processing?!*
– *In many applications, we deal with the same thing!*

$\boxed{\textit{This gets clear in the next section}}$

Starting with Compressive Sensing

Compressive Sensing: *Signal Sampling*



How do we deal with such a continuous-time signal?

- *We first sample it with large enough sampling rate*
- *We recover it from discrete-time samples*

Compressive Sensing: *Signal Sampling*

Shannon Sampling Theorem

Let the Fourier transform of $x(t)$ be non-zero only within $[-B, B]$. Then, the signal is perfectly recovered from samples

$$y_m = x(t_m = mT_s)$$

when $T_s < 1/2B$

How do we do the recovery?

$$x(t) = \sum_{m=-\infty}^{\infty} y_m \operatorname{sinc}\left(\frac{\pi}{T_s} t - m\pi\right)$$

Compressive Sensing: *Signal Sampling*

Shannon Sampling Theorem

Let the Fourier transform of $x(t)$ be non-zero only within $[-B, B]$. Then, the signal is perfectly recovered from samples

$$y_m = x(t_m = mT_s)$$

when $T_s < 1/2B$

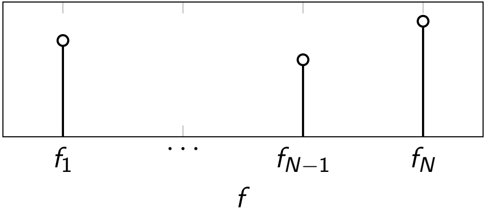
- *In this theorem, what is the signal dimension?* ∞
- *What about the number of samples?* ∞

Compressive Sensing: *Signal Sampling*

How can we go for finite dimension?

Consider the following signal

$$x(t) = \sum_{n=1}^{N} x_n \exp\{2\pi j f_n t\}$$



Compressive Sensing: *Signal Sampling*

From Fourier analysis, we know that

$\boxed{\textit{It covers almost all signals we deal with}}$

- *It consists of N harmonics*
- *Most practical signals consists of finite harmonics*

Compressive Sensing: *Signal Sampling*

What happens if we employ Shannon sampling theorem?

- *The signal is band-limited: Only non-zero between f_1 and f_N*
- *We can sample it with sampling rate faster than $2(f_N - f_1)$*

What happens then?

- *Due to periodicity of the signal, many samples repeat*
- *Keeping the samples in one period, you end up with N samples*

There is however an alternative way!

Compressive Sensing: *Signal Sampling*

Consider the following signal

$$x(t) = \sum_{n=1}^{N} x_n \exp\{2\pi j f_n t\}$$

Let us sample $x(t)$ at a particular time t_m

$$y_m = x(t_m) = \sum_{n=1}^{N} x_n e^{2\pi j f_n t_m} = \underbrace{\left[e^{2\pi j f_1 t_m}, \ldots, e^{2\pi j f_N t_m} \right]}_{\boldsymbol{a}^{\mathsf{T}}(t_m)} \underbrace{\begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}}_{\boldsymbol{x}}$$

Compressive Sensing: *Signal Sampling*

Consider the following signal

$$x(t) = \sum_{n=1}^{N} x_n \exp\{2\pi j f_n t\}$$

Now what happens if we collect M time samples?

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} x(t_1) \\ \vdots \\ x(t_M) \end{bmatrix} = \begin{bmatrix} \boldsymbol{a}^{\mathsf{T}}(t_1) \\ \vdots \\ \boldsymbol{a}^{\mathsf{T}}(t_M) \end{bmatrix} \boldsymbol{x} = \mathbf{A}\, \boldsymbol{x}$$

Compressive Sensing: *Signal Sampling*

Consider the following signal

$$x(t) = \sum_{n=1}^{N} x_n \exp\{2\pi j f_n t\}$$

Now what happens if we collect M time samples?

- *Well, we know this guy $\mathbf{y} = \mathbf{A}\mathbf{x}$: Linear system of equations!*
- *Without side constraints, we need exactly $N = M$ samples!*

How exactly this way meet the sampling theorem?

You will find out in assignments

Compressive Sensing: *Signal Sampling*

In compressive sensing, we deal with signals like

$$x(t) = \sum_{n=1}^{N} x_n \exp\{2\pi j f_n t\}$$

whose $\boldsymbol{x}$ *is sparse*

We can hence take less samples than $M = N$*, or in other words*

compressively sense the signal $x(t)$*,*

and still recover $x(t)$ *perfectly from the samples*

Starting with Compressive Sensing

Section 1: Sparsity and Compressibility

Compressive Sensing: *Signal Sparsity*

An audience might ask now the following question:

Question

Why should $\boldsymbol{x}$ be sparse at all?

In a nutshell, the answer to this question follows two points:

- *Most signals are sparse in transformed bases*
- *Many signals might not be sparse but they are compressible*

We go through these items now!

Compressive Sensing: *Signal Sparsity*

$\boxed{\textit{Most signals are sparse in transformed bases}}$

Many signals are generated from a dictionary

$$\mathcal{D} = \{\phi_1, \ldots, \phi_K\}$$

$\phi_1, \ldots, \phi_K$ *are called atoms and are N-dimensional vectors*

Compressive Sensing: *Signal Sparsity*

Most signals are sparse in transformed bases

Using the dictionary, $\boldsymbol{x}$ is given as

$$\boldsymbol{x} = \sum_{k=1}^{K} c_k \phi_k = [\phi_1, \ldots, \phi_K] \begin{bmatrix} c_1 \\ \vdots \\ c_K \end{bmatrix} = \boldsymbol{\Phi} \boldsymbol{c}$$

From our discussions on a linear system of equations, we know

The maximum number of atoms is N

However, in most applications $K \ll N$

Compressive Sensing: *Signal Sparsity*

Most signals are sparse in transformed bases

It is true that $\boldsymbol{x}$ *is usually not sparse, but*

It is usually presented with only few atoms: $\boldsymbol{c}$ *is sparse*

How shall we use this?

Using the dictionary, we write the signal samples as

$$\boldsymbol{y} = \mathbf{A}\,\boldsymbol{x} = \underbrace{\mathbf{A}\,\boldsymbol{\Phi}}_{\hat{\mathbf{A}}}\,\boldsymbol{c} = \hat{\mathbf{A}}\boldsymbol{c}$$

and we know that $\|\boldsymbol{c}\|_0 \leq s$ *for some* $s \ll K$

Compressive Sensing: *Signal Sparsity*

Most signals are sparse in transformed bases

We use sparse recovery and recover $\boldsymbol{c}$ *by solving*

$$\hat{\mathbf{A}}\boldsymbol{c} = \boldsymbol{y}$$
$$\text{subject to } \|\boldsymbol{c}\|_0 \leq s$$

We then find $\boldsymbol{x}$ *by transforming the bases*

$$\boldsymbol{x} = \boldsymbol{\Phi}\boldsymbol{c}$$

Compressive Sensing: *Signal Sparsity*

> *Many signals might not be sparse but compressible*

In many applications, $\boldsymbol{x}$ is so-called compressible

> *Many entries of $\boldsymbol{x}$ are close to zero*

As the result, we could say that

> *The signal is well-approximated by a sparse signal*

Compressive Sensing: *Signal Sparsity*

Compressibility

$\boldsymbol{x} \in \mathbb{R}^N$ *is compressible, if there exists an s-sparse signal* $\boldsymbol{z} \in \mathbb{R}^N$ *with* $s \ll N$*, such that the error of approximating* $\boldsymbol{x}$ *by* $\boldsymbol{z}$ *is small*

For instance, we want to approximate $\boldsymbol{x}$ *with a* 1*-sparse signal*

$$\boldsymbol{x} = \begin{bmatrix} 0.01 \\ 0.23 \\ 18 \\ 0.06 \end{bmatrix} \qquad \boldsymbol{z}_1 = \begin{bmatrix} \alpha \\ 0 \\ 0 \\ 0 \end{bmatrix} \; \boldsymbol{z}_2 = \begin{bmatrix} 0 \\ \alpha \\ 0 \\ 0 \end{bmatrix} \; \boldsymbol{z}_3 = \begin{bmatrix} 0 \\ 0 \\ \alpha \\ 0 \end{bmatrix} \; \boldsymbol{z}_4 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \alpha \end{bmatrix}$$

We choose $\boldsymbol{z}_3$ *and set* $\alpha = 18$

$\mathbb{R}$ is the real axis

Compressive Sensing: *Signal Sparsity*

But how much we lose when we use the concept of compressibility?

- *To answer, we need to go through some definitions*

Starting with Compressive Sensing

Section 2: Some Basic Definitions

Compressive Sensing: *Some Definitions*

ℓ_p-Norm

Consider vector $\boldsymbol{x}$ with N entries. Its ℓ_p-norm is defined as

$$\|\boldsymbol{x}\|_p = \left(\sum_{n=1}^{N} |x_n|^p \right)^{1/p}$$

Well-known case is the ℓ_2-norm also being called Euclidean norm

$$\|\boldsymbol{x}\|_2 = \sqrt{\sum_{n=1}^{N} |x_n|^2}$$

Compressive Sensing: *Some Definitions*

Few notes on ℓ_p-norm:

- *For $p \geq 1$, the function $f(\boldsymbol{x}) = \|\boldsymbol{x}\|_p$ is convex*

  Local minimum of $\|\boldsymbol{x}\|_p$ in a convex set is a global minimum

- *For $p < 1$, the function $f(\boldsymbol{x}) = \|\boldsymbol{x}\|_p$ is non-convex*

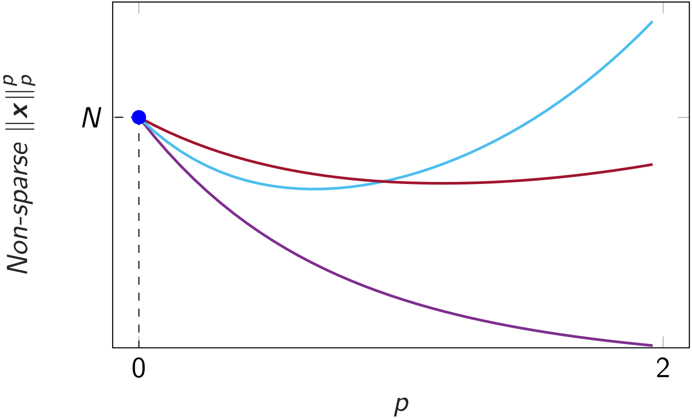
  Minimizing $\|\boldsymbol{x}\|_p$ in this case is a hard task

Why do we care about convexity

You will see it very clearly in next sections

Compressive Sensing: *Some Definitions*

Note that ℓ_p-norm does not change monotonically in terms of p



Compressive Sensing: *Some Definitions*

Attention

A person with background on functional analysis might say

ℓ_p-norm with $p < 1$ is actually not a mathematical norm!

Right! But we don't want confusion! So, we still call it ℓ_p-norm

Compressive Sensing: *Some Definitions*

Unit ℓ_p-Ball

In $\mathbb{R}^N$, the unit ℓ_p-ball is the set of all $\boldsymbol{z} \in \mathbb{R}^N$ such that

$$\|\boldsymbol{z}\|_p \leq 1$$

Let's try an example together:

Let $N = 2$. The unit ℓ_p-ball contains all $\boldsymbol{z} = [z_1, z_2]^\mathsf{T}$

$$\left(|z_1|^p + |z_2|^p\right)^{1/p} \leq 1 \rightsquigarrow |z_1|^p + |z_2|^p \leq 1$$

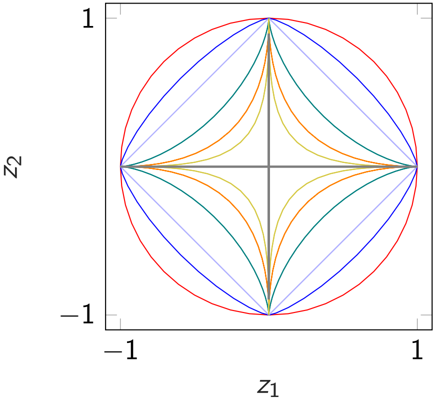
Compressive Sensing: *Some Definitions*

We have

$$\boxed{|z_1|^p + |z_2|^p \leq 1}$$

Let's try

- $p = 2$
- $p = 4/3$
- $p = 1$
- $p = 2/3$
- $p = 1/2$
- $p = 2/5$
- $p \to 0$



Compressive Sensing: *Some Definitions*

We are now ready to answer the question: We first define

ℓ_p-Error of Best s-Sparse Approximation

The ℓ_p-error of the best s-sparse approximation of vector $\boldsymbol{x}$ is

$$\sigma_s\left(\boldsymbol{x}\right)_p = \min_{\boldsymbol{z}} \|\boldsymbol{x} - \boldsymbol{z}\|_p \quad \text{subject to } \|\boldsymbol{z}\|_0 \leq s$$

This metric calculates the error we get when ...

We approximate a compressible signal with its sparse representation

Compressive Sensing: *Some Definitions*

It is easy to bound $\sigma_s(\boldsymbol{x})_p$:

Remember this example: Approximate $\boldsymbol{x}$ with a 1-sparse signal

$$\boldsymbol{x} = \begin{bmatrix} 0.01 \\ 0.23 \\ 18 \\ 0.06 \end{bmatrix} \qquad \boldsymbol{z}_1 = \begin{bmatrix} \alpha \\ 0 \\ 0 \\ 0 \end{bmatrix} \boldsymbol{z}_2 = \begin{bmatrix} 0 \\ \alpha \\ 0 \\ 0 \end{bmatrix} \boldsymbol{z}_3 = \begin{bmatrix} 0 \\ 0 \\ \alpha \\ 0 \end{bmatrix} \boldsymbol{z}_4 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \alpha \end{bmatrix}$$

We choose $\boldsymbol{z}_3$ and set $\alpha = 18$

Compressive Sensing: *Some Definitions*

We can always find best s-sparse approximation, in the same way

- *Sort the entries of* $\boldsymbol{x}$ *as*

$$|x_{i_1}| \geq |x_{i_2}| \geq \ldots \geq |x_{i_N}|$$

$$\boldsymbol{x} = \begin{bmatrix} 0.01 \\ 0.23 \\ 18 \\ 0.06 \end{bmatrix} \qquad i_1 = 3, i_2 = 2, i_3 = 4, i_4 = 1$$

Compressive Sensing: *Some Definitions*

We can always find best s-sparse approximation, in the same way

- *Set* $\mathbf{z}^\star$ *to be s-sparse with support*

$$\mathrm{Supp}\left(\mathbf{z}^\star\right) = \{i_1, \ldots, i_s\}$$

whose non-zero entries are

$$z^\star_{i_1} = x_{i_1}, \ldots, z^\star_{i_s} = x_{i_s}$$

Compressive Sensing: *Some Definitions*

$\boxed{\boldsymbol{z}^\star \text{ is the best } s\text{-sparse approximation of vector } \boldsymbol{x}}$

Therefore, the ℓ_p-error is given by

ℓ_p-Error of Best s-Sparse Approximation

The ℓ_p-error of the best s-sparse approximation of vector $\boldsymbol{x}$ is

$$\sigma_s(\boldsymbol{x})_p = \min_{\boldsymbol{z}} \|\boldsymbol{x} - \boldsymbol{z}\|_p = \|\boldsymbol{x} - \boldsymbol{z}^\star\|_p$$

$$\text{subject to } \|\boldsymbol{z}\|_0 \leq s$$

Compressive Sensing: *Some Definitions*

$\sigma_s(\boldsymbol{x})_p^p$ *in this case is*

$$
\begin{aligned}
\sigma_s(\boldsymbol{x})_p^p = \|\boldsymbol{x} - \boldsymbol{z}^\star\|_p^p &= \sum_{n=1}^{N} |x_{i_n} - z_{i_n}^\star|^p \\
&= \sum_{n=1}^{s} |x_{i_n} - z_{i_n}^\star|^p + \sum_{n=s+1}^{N} |x_{i_n} - z_{i_n}^\star|^p \\
&= \sum_{n=1}^{s} |x_{i_n} - x_{i_n}|^p + \sum_{n=s+1}^{N} |x_{i_n} - 0|^p \\
&= \sum_{n=s+1}^{N} |x_{i_n}|^p
\end{aligned}
$$

Compressive Sensing: *Some Definitions*

Assume $q < p$, now we write

$$\sigma_s(\boldsymbol{x})_p^p = \sum_{n=s+1}^{N} |x_{i_n}|^p = \sum_{n=s+1}^{N} |x_{i_n}|^{p-q} |x_{i_n}|^q$$

Since $|x_{i_s}| \geq |x_{i_n}|$ for $n = s+1, \ldots, N$, we have

$$\begin{aligned} \sigma_s(\boldsymbol{x})_p^p &\leq \sum_{n=s+1}^{N} |x_{i_s}|^{p-q} |x_{i_n}|^q \\ &= |x_{i_s}|^{p-q} \sum_{n=s+1}^{N} |x_{i_n}|^q \end{aligned}$$

Compressive Sensing: *Some Definitions*

We can further rewrite the inequality as

$$\sigma_s(\boldsymbol{x})_p^p \leq |x_{i_s}|^{p-q} \sum_{n=s+1}^{N} |x_{i_n}|^q$$
$$= \left(|x_{i_s}|^q\right)^{\frac{p-q}{q}} \sum_{n=s+1}^{N} |x_{i_n}|^q$$

Since $|x_{i_s}| \leq |x_{i_n}|$ *for* $n = 1, \ldots, s$, *we can always write*

$$|x_{i_s}|^q = \frac{1}{s}\sum_{n=1}^{s}|x_{i_s}|^q \leq \frac{1}{s}\sum_{n=1}^{s}|x_{i_n}|^q$$

Compressive Sensing: *Some Definitions*

Therefore, we have

$$
\begin{aligned}
\sigma_s(\boldsymbol{x})_p^p &\leq \left(|x_{i_s}|^q\right)^{\frac{p-q}{q}} \sum_{n=s+1}^{N} |x_{i_n}|^q \\
&\leq \left(\frac{1}{s}\sum_{n=1}^{s}|x_{i_n}|^q\right)^{\frac{p-q}{q}} \sum_{n=s+1}^{N} |x_{i_n}|^q \\
&\leq \left(\frac{1}{s}\|\boldsymbol{x}\|_q^q\right)^{\frac{p-q}{q}} \|\boldsymbol{x}\|_q^q \\
&= s^{\frac{q-p}{q}} \|\boldsymbol{x}\|_q^p
\end{aligned}
$$

Compressive Sensing: *Some Definitions*

We concluded that

$$\sigma_s\left(\boldsymbol{x}\right)_p^p \leq s^{\frac{q-p}{q}} \left\|\boldsymbol{x}\right\|_q^p$$

Or equivalently

$$\sigma_s\left(\boldsymbol{x}\right)_p \leq s^{\frac{1}{p}-\frac{1}{q}} \left\|\boldsymbol{x}\right\|_q$$

Compressive Sensing: *Some Definitions*

What does this result say?

Let $p = 2$ and $q = 1$; then, this bound says

$$\sigma_s(\boldsymbol{x})_2 \leq \frac{\|\boldsymbol{x}\|_1}{\sqrt{s}}$$

If the signal has a bounded ℓ_1-norm

As s grows large, the ℓ_2-error drops

There exists an $s < N$ at which the error becomes negligible

Compressive Sensing: *Some Definitions*

Moral of Story

- *Most signals represented with finite numbers of entries*
- *These entries are presented by a vector which is*

  Either sparse or compressible
- *We could recover the signal perfectly from few samples*

Final Points

Typical Questions Coming to Your Mind

You now probably come up with various questions

- *How much we could reduce the number of samples?*
- *How could we recover the signal efficiently from the samples?*
- *How should we perform the sampling?*
- *Can we guarantee the perfect recovery of a signal?*

$\boxed{\textit{No worries! These are what we learn in the first part}}$

Which Parts of Textbooks?

We are now over with the introductory part!

I would suggest to go over the text book

A Mathematical Introduction to Compressive Sensing
S. Foucart and H. Rauhut, Book, 2013

and study the following parts:

- *Chapter 1: Sections 1.1 and 1.2, Pages 1–23*
- *Chapter 2: Section 2.1, Pages 41–47*